

(19)



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **08063483 A**(43) Date of publication of application: **08.03.96**

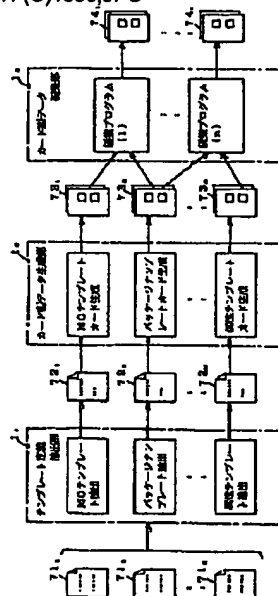
(51) Int. Cl.

**G06F 17/30****G06F 17/27**(21) Application number: **06194988**(71) Applicant: **FUJITSU LTD**(22) Date of filing: **19.08.94**(72) Inventor: **FUKUDA KENICHI**(54) **INFORMATION ANALYSIS AND EDITING SYSTEM** COPYRIGHT: (C)1996,JPO

(57) Abstract:

PURPOSE: To easily perform editing corresponding to a purpose by extracting template data on a part to be read from text data, made in syntax with reserved words, through syntax analysis, and recording the contents of the extracted template data in corresponding fields of card type data having fixed-form item fields.

CONSTITUTION: A template definition extraction part  $1_1$  extracts template data  $72_1-72_m$  on the part to be read from the text data  $71_1-71_k$  made in syntax with the reserved words. A card type data generation part 12 records the contents of the extracted template data in the corresponding fields of the card type data  $72_1-72_m$  having fixed-form item fields. A card type data editing part  $1_3$  inputs the card type data  $73_1-73_m$ , retrieves the card type data by using the contents of specific card type data as a retrieval key, and gathers information on the corresponding item fields from one or  $\geq 2$  retrieved card type data to generate new card type data  $74_1-74_n$ .

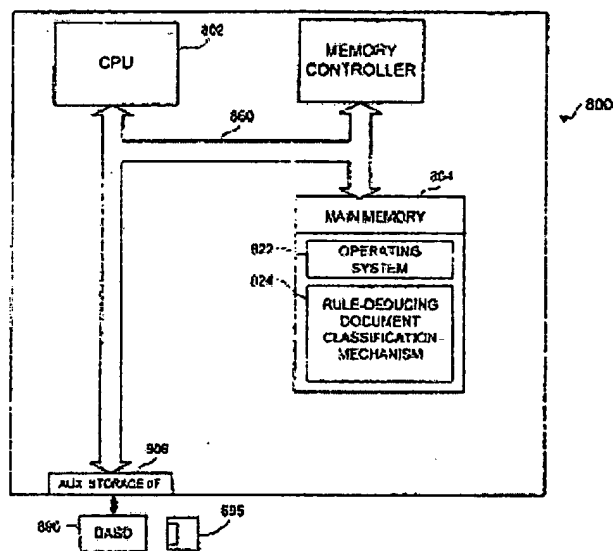


## Method for automatic deduction of rules for matching content to categories

**Patent number:** US2004139059  
**Publication date:** 2004-07-15  
**Inventor:** CONROY WILLIAM F [US]; GOSBY DESIREE D G [US]  
**Applicant:**  
**Classification:**  
- international: G06F7/00  
- european: G06F17/30D  
**Application number:** US20020335351 20021231  
**Priority number(s):** US20020335351 20021231

### Abstract of US2004139059

Accordingly, the invention is a method for automatic deduction of rules for matching document content to a category within a strange taxonomy, which allows the document to be automatically classified into a proper category for storage in that strange taxonomy. The method includes the steps of spidering the taxonomy to determine its structure and contents, extracting keywords from documents within the strange taxonomy, formulating rules for determining the category from the extracted keywords, and applying the rules to classify a new document whose keywords have been extracted. The taxonomy is strange because the user has no knowledge of its internal structure and needs no such knowledge. The taxonomy may be flat or may be hierarchal, the later having rules formulated at each level for proceeding to the next level. Variations for creating new and refurbishing old document management systems are disclosed.



Data supplied from the *esp@cenet* database - Worldwide

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-63483

(43)公開日 平成8年(1996)3月8日

(51)Int.Cl.*	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 F 17/30 17/27		9194-5L 8420-5L	G 0 6 F 15/ 40 15/ 38	3 7 0 A J

審査請求 未請求 請求項の数4 O L (全 11 頁)

(21)出願番号 特願平6-194988

(22)出願日 平成6年(1994)8月19日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72)発明者 福田 健一

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

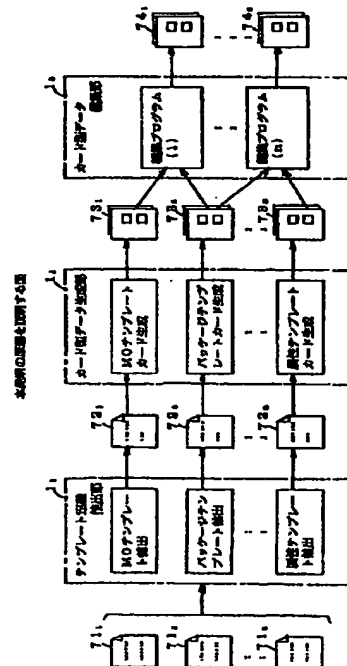
(74)代理人 弁理士 井桁 貞一

(54)【発明の名称】 情報解析・編集システム

(57)【要約】 (修正有)

【目的】 予約語により構文化されているテキストデータから所望の情報を抽出し、目的に応じた編集を容易に行えるようにする。

【構成】 予約語により構文化されているテキストデータ71から該予約語を利用した構文解析により読みたい部分のテンプレートデータ72を抽出するテンプレート定義抽出部11と、抽出したテンプレートデータ72の内容を定型の項目欄を有するカード型データ73の対応欄に夫々記録するカード型データ生成部12とを備える。カード型データ生成部12は各テンプレートデータをテンプレート別のカード型データ731～73nに夫々記録する。また、上記カード型データ731～73nを入力とし、かつ所定のカード型データの内容を検索キーとして該カード型データを検索すると共に、検索の得られた1又は2以上のカード型データから対応する項目欄の情報を収集することで新たなカード型データ74を生成するカード型データ編集部13を備える。



## 【特許請求の範囲】

【請求項1】 予約語により構文化されているテキストデータから該予約語を利用した構文解析により読みたい部分のテンプレートデータを抽出するテンプレート定義抽出部と、  
前記抽出したテンプレートデータの内容を定型の項目欄を有するカード型データの対応欄に夫々記録するカード型データ生成部とを備えることを特徴とする情報解析・編集システム。

【請求項2】 テキストデータはGDMO定義文から成ることを特徴とする請求項1の情報解析・編集システム。

【請求項3】 テンプレート定義抽出部は複数種のテンプレートデータを抽出し、かつカード型データ生成部は前記抽出された各テンプレートデータをテンプレート別のカード型データに夫々記録することを特徴とする請求項2の情報解析・編集システム。

【請求項4】 請求項3の複数のカード型データを入力とし、かつ所定のカード型データの内容を検索キーとして該カード型データを検索すると共に、検索の得られた1又は2以上のカード型データから対応する項目欄の情報を収集することで新たなカード型データを生成するカード型データ編集部を備えることを特徴とする情報解析・編集システム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は情報解析・編集システムに関し、更に詳しくは予約語により構文化されているテキストデータから所望の情報を抽出し、編集を行う情報解析・編集システムに関する。ISO国際標準のOSIネットワーク管理又はITU-T勧告のTMNに基づくネットワーク（被）管理システムの開発においては、ISO、ITU-T、その他の団体により作成されたGDMO定義に準拠していることが要求される。このため、インタフェースの仕様化やシステム設計を行う者は膨大な量のGDMO定義を読んで理解し、網羅的な（国際）標準のGDMO定義から装置にインプリメントする範囲を選択する必要がある。

## 【0002】

【従来の技術】図11～図13は従来の印刷物によるGDMO定義の一部を示す図（1）～（3）である。図11において、例えばMOクラスにおける「equipment」の定義の内容を知りたい場合は、項目3.2.1の「equipment MANAGED OBJECTCLASS」の欄を読む。ここで、大文字で示す「MANAGED OBJECTCLASS」はMOクラスを表す予約語であり、小文字で示す「equipment」はこの定義文で使用する管理対象物の名前である。

【0003】更に、次段の予約後「DERIVED FROM」の欄をその構文の一区切り「;」まで読むと、「equipment」についてのスーパークラスは「top」により定義さ

れており、該「top」についての定義は他のページにある。更に次の予約後「CHARACTERIZED BY」により関連する他の詳細情報が示されている。例えば、予約後「PACKAGE」の対象名は「equipmentPackage」であり、その属性（:ATTRIBUTES）は、  
equipmentId（注：#1、#2等と番号付けされる）  
replaceable（注：置き換え可能）  
である。

【0004】なお、「equipmentPackage」については、「属性」の他に「アクション」及び「通知（:NOTIFICATIONS）」についてもセットで知りたいのであるが、この記載例では他のページを参照する必要がある。一方、動作の注釈（:BEHAVIOUR）は図の位置に埋め込まれている。しかし、「BEHAVIOUR」は他のかけ離れたページに書かれている場合も少なくない。

【0005】図12において、更にMOクラス「equipment」の条件パッケージの予約語「CONDITIONAL PACKAGES」の欄には「createDeleteNotificationsPackage」を始めとする各種条件パッケージの名前が記載されている。そして、これらの各条件パッケージの内容（属性、アクション、通知等）を知りたい場合は他のページを参照する必要がある。

【0006】因みに、図13の項目4.10には「createDeleteNotificationsPackage」の「通知」について定義されている。即ち、

createDeleteNotificationsPackage PACKAGE  
NOTIFICATIONS

objectCreation（注：データ作成通知）

objectDeletion（注：データ削除通知）

である。

【0007】従来は、人手により、まず印刷物の目次や索引をたよりにGDMO定義間の参照関係をたどり、GDMO定義の意味を理解していた。しかる後、GDMO定義から装置にインプリメントする範囲を選択していた。

## 【0008】

【発明が解決しようとする課題】しかし、現実には、

- ① 膨大な量のGDMO定義が作成されている
- ② 各MO定義は多段構成をとり、かつ分散している（例えば「BEHAVIOUR」が飛び離れている）
- ③ MO定義間には複雑かつ大量の参照関係がある
- ④ MOの正確な意味づけは他のMOとの関係によって間接的に与えられる
- ⑤ MO定義はインプリメント範囲の選択に適した形式とはなっておらず、その読解作業には大変時間がかかり、インプリメント範囲の選択ができるようになるまでには大変な記憶力を要する作業となる、等の状況にある。

【0009】このため、従来は、

- ⑥ GDMO定義は大変理解しにくい

⑦ インプリメント範囲の選択は極めて困難

⑧ GDMO定義が修正されると、それに伴ってインタフェース仕様を手作業で修正しなければならない、等の様々な問題が生じていた。

【0010】なお、エディタやワープロ等の文書処理装置の利用も考えられるが、これらの装置では簡単な文字列の検索しかできず、例えばMO定義のタイトルである「MANAGED OBJECT CLASS network」とMO定義内の注釈欄に現れる「MANAGED OBJECT CLASS network is an object which …」とを区別なく探し出してしまふ。本発明の目的は、予約語により構文化されているテキストデータから所望の情報を抽出し、目的に応じた編集を容易に行える情報解析・編集システムを提供することにある。

【0011】

【課題を解決するための手段】上記の課題は図1の構成により解決される。即ち、本発明(1)の情報解析・編集システムは、予約語により構文化されているテキストデータから該予約語を利用した構文解析により読みたい部分のテンプレートデータを抽出するテンプレート定義抽出部と、前記抽出したテンプレートデータの内容を定型の項目欄を有するカード型データの対応欄に夫々記録するカード型データ生成部とを備えるものである。

【0012】また本発明(4)の情報解析・編集システムは、請求項3の複数のカード型データを入力とし、かつ所定のカード型データの内容を検索キーとして該カード型データを検索すると共に、検索の得られた1又は2以上のカード型データから対応する項目欄の情報を収集することで新たなカード型データを生成するカード型データ編集部を備えるものである。

【0013】

【作用】本発明(1)の情報解析・編集システムにおいて、テンプレート定義抽出部1は予約語により構文化されているテキストデータ71から該予約語を利用した構文解析により読みたい部分のテンプレートデータ72を抽出する。そして、カード型データ生成部1は前記抽出したテンプレートデータの内容を定型の項目欄を有するカード型データ73の対応欄に夫々記録する。

【0014】従って本発明(1)によれば、複雑な文書構造を有するテキストデータ71の中の読みたい部分のテンプレートデータが定型の項目欄を有するカード型データ73に変換されて出力されるので、必要な情報の収集を自動化できると共に、情報検索に適したデータベースを容易に構築できる。好ましくは、テキストデータはGDMO定義文から成っている。

【0015】従って、複雑なGDMO定義の検索や参照がコンピュータ処理可能となり、ネットワーク(被)管理システムのインタフェース仕様化の作業の低減に寄与するところが大きい。また好ましくは、テンプレート定義抽出部1は複数種のテンプレートデータ72<sub>1</sub>～72<sub>n</sub>を抽出し、かつカード型データ生成部1は前記抽出された各テンプレートデータ72<sub>1</sub>～72<sub>n</sub>をテンプレート別のカード型データ73<sub>1</sub>～73<sub>n</sub>に夫々記録する。

【0016】従って、例えばGDMO定義に見られるように各テンプレートデータ72<sub>1</sub>～72<sub>n</sub>が夫々にユニークな構造をしていても、効率のよいデータベースを構築できる。また本発明(4)の情報解析・編集システムにおいて、カード型データ編集部1は請求項3の複数のカード型データ73を入力とし、かつ所定のカード型データの内容を検索キーとして該カード型データを検索すると共に、検索の得られた1又は2以上のカード型データから対応する項目欄の情報を収集することで新たなカード型データ74を生成する。

【0017】従って本発明(4)によれば、請求項3で得られた例えばGDMO定義のデータベースを基に、元のテキストデータにある様々な参照関係をたどり、詳細データの一覧リストを作成することが可能であり、これを見れば、装置にインプリメントする範囲を容易に理解し、かつ容易に取捨選択できる。

【0018】

【実施例】以下、添付図面に従って本発明による実施例を詳細に説明する。なお、全図を通して同一符号は同一又は相当部分を示すものとする。図2は実施例の情報解析・編集システムのブロック図で、図において1はCPU、2はCPU1が実行する図3～図5の処理プログラムを記憶しているROM、3はディスプレイ(DSP)、4はディスク装置(DSK)、5はキーボード(KBD)、6はマウス等のポインティングデバイス(PD)、7はRAM、71はテキストデータの記憶エリア、72はテンプレート定義データの記憶エリア、73はカード型データの記憶エリア、74は編集結果の記憶エリア、75はその他のエリア、8はプリンタ(PRN)、9はCPU1の共通バスである。

【0019】CPU1はROM2と共に図1のテンプレート定義抽出部1、カード型データ生成部1及びカード型データ編集部1の諸機能を実現する。RAM7は図1のテキストデータ71<sub>1</sub>～71<sub>n</sub>、テンプレート定義データ72<sub>1</sub>～72<sub>n</sub>、カード型データ73<sub>1</sub>～73<sub>n</sub>、編集結果のデータ74<sub>1</sub>～74<sub>n</sub>の夫々一部を記憶し、残りはディスク装置34のデータベースに格納される。

【0020】図3は実施例のテンプレート定義抽出処理のフローチャートである。テンプレート定義抽出処理ではGDMOテキストデータを入力し、予約語を利用した構文解析により読みたい部分のテキストデータ(テンプレート定義データ)を切り出す。GDMOテキストデータには通信網関連のテキストデータ71<sub>1</sub>、コンピュータネットワーク関連のテキストデータ71<sub>2</sub>等の各種テキストデータが含まれている。テキストデータは図11

に示す如く予約語を使用して構文化されている。但し、項目番号と共に記載されているところの、

### 「3. 2 Managed Element Fragment

Managed object classes in ... as follows:

#### 3. 2. 1 Equipment」

等の欄については印刷物に固有の欄であり、GDMOテキストデータには含まれていない。

【0021】ステップS1ではGDMOテキストデータを入力する。ステップS2ではひとかたまりのテンプレート定義文(例えば、図11～図12の項目3. 2. 1に相当する欄)を読み込む。ステップS3では頭から予約語を調べ、指定種別(即ち、読みたい部分のテンプレート定義)か否かを判別する。指定種別としてはMO (MANAGED OBJECTCLASS)、パッケージ(PACKAGE, CONDITIONALPACKAGES)、属性(ATTRIBUTES)、アクション(ACTIONS)、ビヘビア(BEHAVIOUR)、通知(NOTIFICATIONS)等がある。

【0022】指定種別の場合はステップS4でテキストデータの予約語、記号「:」、定義文の階層構造等に基づき、内容を構文解析し、キーワード(対象名等)毎に区切記号を入れて一つのテンプレート定義データを抽出する。また指定種別でない場合はステップS4の処理をスキップする。ステップS5では上記のひとかたまりのテンプレート定義文の中に埋め込まれている他のテンプレート定義があるか否かを調べ、ある場合はステップS6で指定種別か否かを判別する。指定種別ならステップS7で内容を構文解析し、区切記号を入れて新たにテンプレート定義データを抽出する。また指定種別でない場合はステップS7の処理をスキップする。

【0023】こうして全ての埋込定義を調べ、必要なテンプレート定義の抽出が終わると、ステップS8で他のひとかたまりのテンプレート定義があるか否かを調べる。ある場合はステップS2に戻り、上記の処理を繰り返す。また無い場合はこの処理を抜ける。こうして、図1のMOから属性に至る各種の複数のテンプレート定義データ72<sub>1</sub>～72<sub>2</sub>が別個に抽出される。

【0024】図4は実施例のカード型データ生成処理のフローチャートである。カード型データ生成処理では上記抽出した各種テンプレート定義データの内容(対象名等)を定型のカード型データの各対応する項目欄に書き込む。図6にMOテンプレートに対応するカード型データの例を示す。このカード型データにはMOクラス、スーパークラス、パッケージ、条件パッケージ等の定型の項目欄(書込フィールド)が設けられている。同様にして、図7はパッケージテンプレート、図8はビヘビアテンプレートに夫々対応するカード型データの例を示しており、書込フィールドは夫々異なっている。

【0025】ステップS11では一つのテンプレート定義データを読み込む。ステップS12では対応するカード型データ(ブランクのもの)を作成する。ステップS

13ではカード型データの書込フィールドを選択し、ステップS14では当該書込フィールドにテンプレート定義の対応箇所のデータ(対象名等)を書き込む。ステップS15では全ての書込フィールドに対して上記の処理を行う。そして、ステップS16では全てのテンプレート定義データに対して上記の処理を行う。こうして、図1のテンプレート別の複数のカード型データ73<sub>1</sub>～73<sub>3</sub>が形成される。

【0026】図6の(A)は図11のMOクラス「equipment」をカード型データに展開した場合の表示例を示しており、スーパークラスの書込フィールドには「top」が、また条件パッケージの書込フィールドには図12の各種条件パッケージの対象名が書き込まれている。同様にして形成された他のカード型データ(MOクラス「broadbandATMSwitchingElement」)の例を図6の(B)に示す。

【0027】なお、枠内の見えない部分は右側の矢印をクリックすれば、画面にスクロール表示される。また、カードの右上にはこのカードが全MOクラスカード298枚中の39枚目であることが示されている。また、このMOクラスのカードにはパッケージテンプレート定義内に記述されている属性名、アクション名、通知名等が現れていないが、それらの定義情報については図7の(A)、(B)に示す如く他にパッケージテンプレート対応のカードが生成されており、そちらに格納されている。また、図8にはビヘビアテンプレート対応のカードの例を示す。

【0028】図5は実施例のカード型データ編集処理のフローチャートである。例えばカード型データ編集処理(1)では、MOクラステンプレート定義対応のカード型データ73<sub>1</sub>とパッケージテンプレート定義対応のカード型データ73<sub>2</sub>とを入力とし、かつ所定のパッケージ名を検索キーにしてカード間の参照関係を調べ、両者を結合したカード型データ74<sub>1</sub>を新たに生成している。

【0029】即ち、ステップS21では例えばRAM7上のその他のエリア75に設けたリスト(LST)1の欄にMOクラスカードのスーパークラスフィールドの内容を読み込む。ステップS22では同様にしてLST2にMOクラスカードのパッケージフィールドの内容を読み込む。ステップS23ではLST2からパッケージ名を一つ取り出し、これを検索キーにしてパッケージテンプレート対応のカード型データ73<sub>2</sub>を検索する。そして、検索が得られると、ステップS24では同じくRAM7に設けたリスト(Lst)1に当該パッケージカードの属性フィールド、Lst2にアクションフィールド、Lst3に通知フィールドの内容を夫々読み込む。ステップS25ではLST2の全てのパッケージ名に対して上記の処理を繰り返す。

【0030】ステップS26では更にRAM7のLST

7

3にMOクラスカードの条件パッケージフィールドの内容を読み込む。ステップS27ではLST3からパッケージ名を一つ取り出し、これを検索キーにしてパッケージテンプレート対応のカード型データ73<sub>i</sub>を検索する。そして、検索が得られると、ステップS28では上記と同様にしてRAM7のlst1に当該パッケージカードの属性フィールド、lst2にアクションフィールド、lst3に通知フィールドの内容を夫々読み込む(連結する)。ステップS29ではLST3の全てのパッケージ名に対して上記の処理を繰り返す。

【0031】ステップS30ではスーパークラス、属性、アクション、通知の各フィールドを有するカード型データ74<sub>i</sub>を新規に作成する。ステップS31ではRAM7のLST1の内容をカード型データ74<sub>i</sub>のスーパークラスフィールドに、lst1の内容を属性フィールドに、lst2の内容をアクションフィールドに、そして、lst3の内容を通知フィールドに夫々書き込む。

【0032】図9に編集結果のカード型データの一例を示す。カード型データ編集処理(1)により生成されたカード型データ74<sub>i</sub>は、MOクラス名と、そのクラスに付属している属性、アクション、通知とを整理された分かり易い関係で表しており、GDMO定義を説明しようとしている作業員に対して整理された意味のある情報を提供している。

【0033】図10は図9のカードを入力とし、かつそのスーパークラス名を検索キーにしてカード型データ73を検索し、複数のカードの属性、アクション、通知を夫々結合した結果を新たなカード型データ74<sub>j</sub>として生成したものである。このカード型データ74<sub>j</sub>は、MOクラスのインスタンスが生成されたときに、どのような属性、アクション、通知をもたねばならないかを示している。

【0034】このように、実施例のカード型データ73(場合によっては74)を利用することで、目的に応じた様々なデータ編集が行え、これにより装置の設計者は、図9又は図10のカード型データ74の内容を取捨選択することで装置のインプリメント範囲を容易に決定できる。なお、上記実施例ではGDMO定義のテキストデータを入力としたが、本発明は予約語により構文化されている他の様々なテキストデータに対しても同様に適用できる。

【0035】また、上記本発明に好適なる実施例を述べたが、本発明思想を逸脱しない範囲内で、構成及びプログラム処理の様々な変更が行えることは言うまでも無い。

【0036】

8

【発明の効果】以上述べた如く本発明(1)によれば、複雑な文書構造を有するテキストデータから必要な情報の自動的に収集できると共に、情報検索に適したデータベースを容易に構築できる。また本発明(4)によれば、上記データベースを基に、元のテキストデータにある様々な参照関係をたどり、詳細データの一覧リストを作成することが可能であり、これを見れば、装置にインプリメントする範囲を容易に理解し、かつ容易に取捨選択できる。

10 【0037】従って、例えば本システムをGDMO定義に提供すれば、ネットワーク(被)管理システムのインタフェース仕様化作業の低減に寄与するところが大きい。

【図面の簡単な説明】

【図1】図1は本発明の原理を説明する図である。

【図2】図2は実施例の情報解析・編集システムのブロック図である。

【図3】図3は実施例のテンプレート定義抽出処理のフローチャートである。

20 【図4】図4は実施例のカード型データ生成処理のフローチャートである。

【図5】図5は実施例のカード型データ編集処理のフローチャートである。

【図6】図6は実施例のカード型データを示す図(1)である。

【図7】図7は実施例のカード型データを示す図(2)である。

【図8】図8は実施例のカード型データを示す図(3)である。

30 【図9】図9は実施例の編集結果のカード型データを示す図(1)である。

【図10】図10は実施例の編集結果のカード型データを示す図(2)である。

【図11】図11はGDMO定義の一部を示す図(1)である。

【図12】図12はGDMO定義の一部を示す図(2)である。

【図13】図13はGDMO定義の一部を示す図(3)である。

40 【符号の説明】

1<sub>1</sub> テンプレート定義抽出部

1<sub>2</sub> カード型データ生成部

1<sub>3</sub> カード型データ編集部

71<sub>i</sub> ~ 71<sub>k</sub> テキストデータ

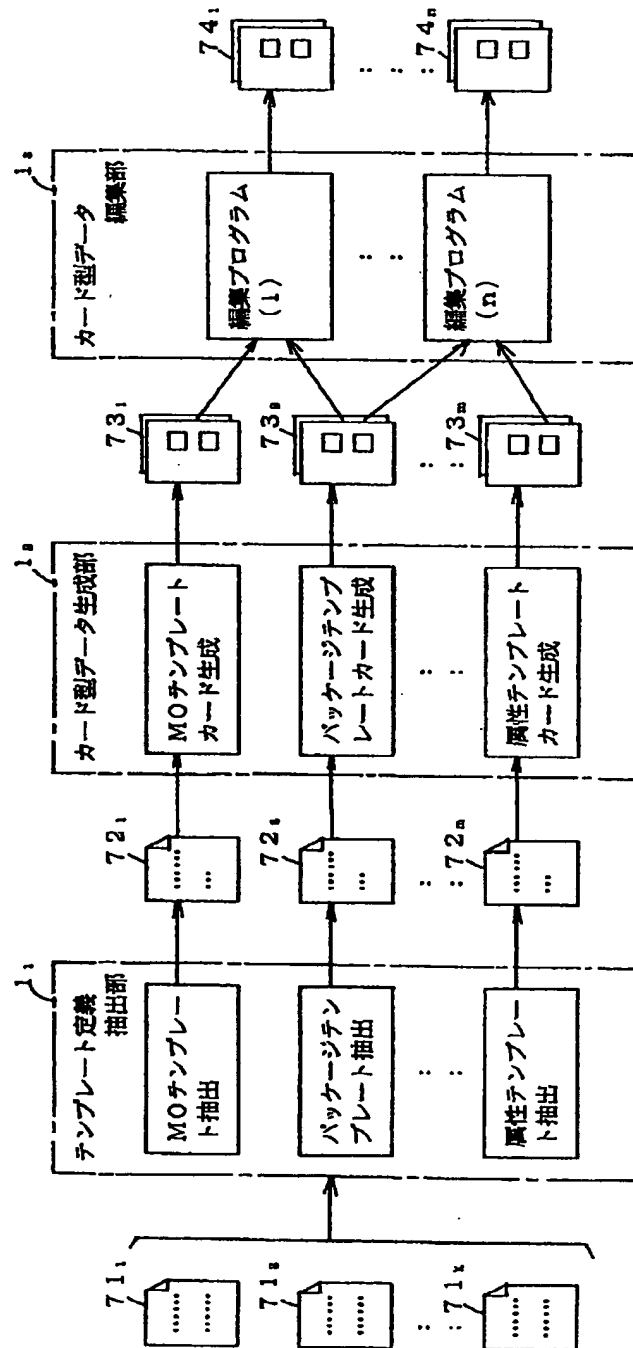
72<sub>i</sub> ~ 72<sub>n</sub> テンプレート定義データ

73<sub>i</sub> ~ 73<sub>n</sub> カード型データ

74<sub>i</sub> ~ 74<sub>n</sub> 編集結果のデータ

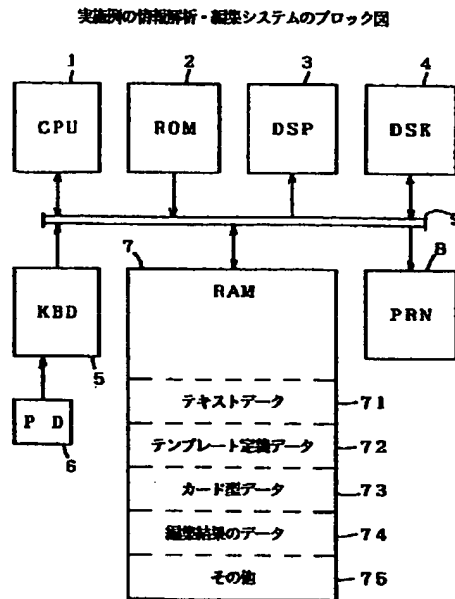
【図1】

本発明の原理を説明する図



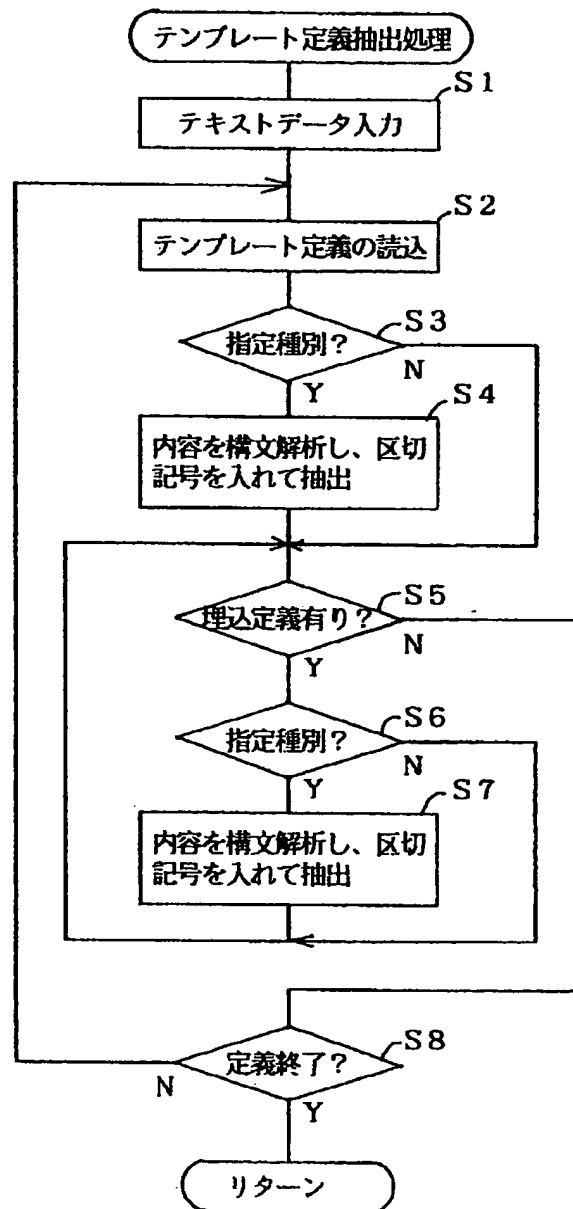


【図2】



【図3】

実施例のテンプレート定義抽出処理のフローチャート



【図13】

GDMO定義の一部を示す図 (3)

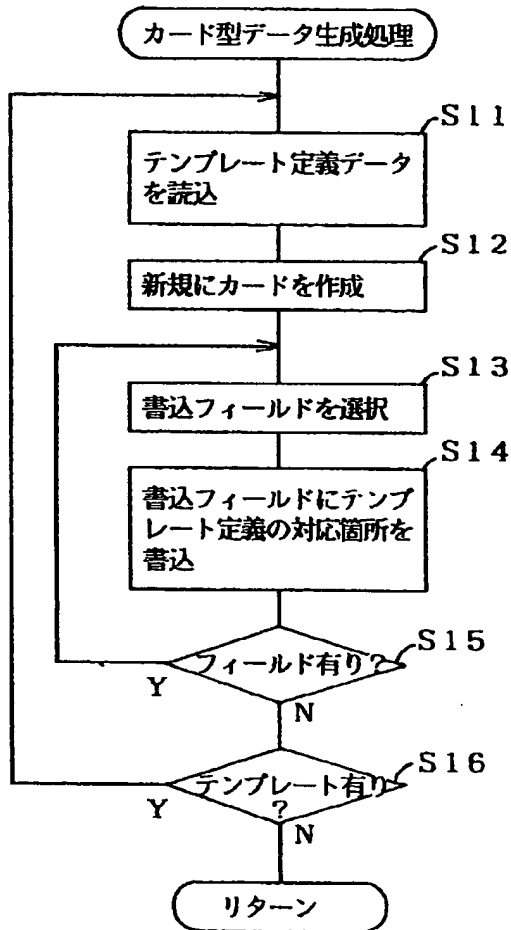
```

ATTRIBUTE
  classNumber GET;
REGISTERED AS {a3100Package 6};
4.7 Characteristic Information
  characteristicInformation PACKAGE
  ATTRIBUTE
    characteristicInformation GET;
REGISTERED AS {a3100Package 7};
4.8 Client Connection
  clientConnection PACKAGE
  ATTRIBUTE
    clientConnection GET;
REGISTERED AS {a3100Package 8};
4.9 Client Trail
  clientTrail PACKAGE
  ATTRIBUTE
    clientTrail GET;
REGISTERED AS {a3100Package 9};
4.10 Create Delete Notifications
  createDeleteNotifications PACKAGE
  NOTIFICATION
    createDeleteNotifications X.721:1992PackageCreate,
    createDeleteNotifications X.721:1992PackageDelete;
REGISTERED AS {a3100Package 10};
4.11 Cross Connection Pointer
  crossConnectionPointer PACKAGE
  ATTRIBUTE
    crossConnectionPointer GET;
REGISTERED AS {a3100Package 11};
4.12 CTF Instance
  ctfInstance PACKAGE
  ATTRIBUTE
    ctfInstance GET;
REGISTERED AS {a3100Package 12};

```

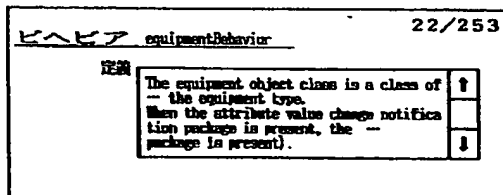
【図4】

実施例のカード型データ生成処理のフローチャート



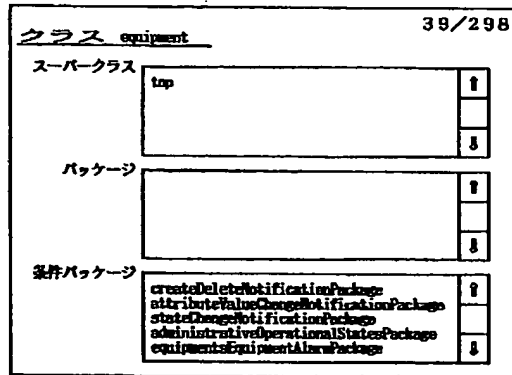
【図8】

実施例のカード型データを示す図(3)

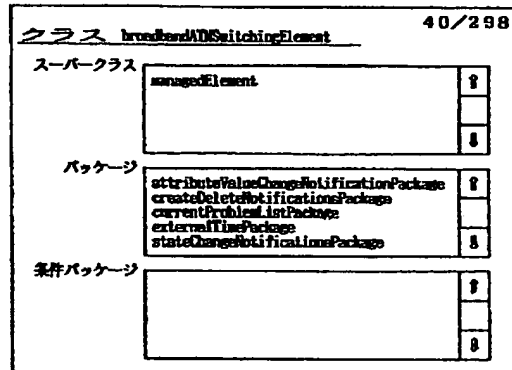


【図6】

実施例のカード型データを示す図(1)



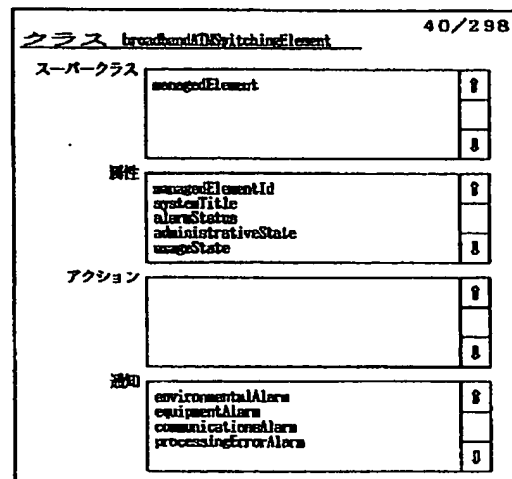
(A)



(B)

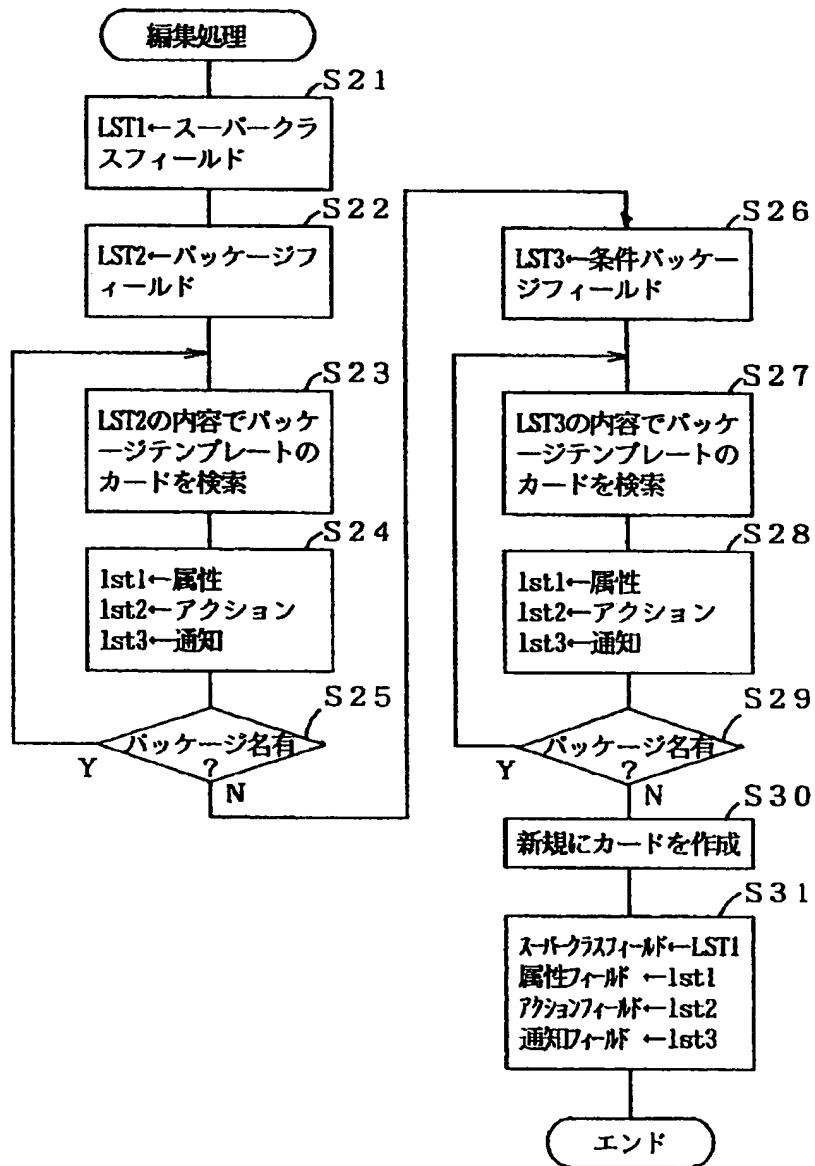
【図9】

実施例の観測結果のカード型データを示す図(1)



【図5】

実施例のカード型データ編集処理のフローチャート



【図7】

実施例のカード型データを示す図(2)

パッケージ equipmentPackage		332/638
属性	equipmentId replacable	↑ ↓
アクション		↑ ↓
通知		↑ ↓

(A)

パッケージ managedElementPackage		333/638
属性	managedElementId systemTitle alarmStatus administrativeState usageState	↑ ↓
アクション		↑ ↓
通知	environmentalAlarm equipmentAlarm communicationsAlarm processingErrorAlarm	↑ ↓

(B)

【図10】

実施例の編成結果のカード型データを示す図(2)

クラス broadbandSwitchingElement		6/16
属性	operationalState usageState managedElementId systemTitle alarmStatus administrativeState usageState userLabel (packageName) (allowmorph)	↑ ↓
アクション	(allowtestableVisualLocalAlarm) (inhibit testableVisualLocalAlarm) (reset testableAlarm)	↑ ↓
通知	environmentalAlarm equipmentAlarm communicationsAlarm processingErrorAlarm attributeValueChange	↑ ↓

注: ( ) はオプション

【図11】

GDMO定義の一部を示す図(1)

network MANAGED OBJECT CLASS  
DERIVED FROM "Recommendation X.721: 1992" msg;  
CHARACTERIZED BY  
networkPackage PACKAGE  
BEHAVIOUR  
networkDefinitionMsg  
ATTRIBUTES  
network GET;  
CONDITIONAL PACKAGES  
networkPackage PRESENT IF "an instance supports it";  
REGISTERED AS (networkClass 1);

networkDefinition BEHAVIOUR  
DEFINED AS "The network object class is a class of managed objects that are collections of interconnected telecommunication and management objects (logical or physical) capable of exchanging information. These objects have one or more common characteristics, for example they may be owned by a single customer or provider, or connected with a specific service network. A network may be used within another (higher) network, thereby forming a containment relationship. An example of a network that is connected to another network is a transmission sub-network. It is owned by a single administration and can only perform transmission functions."

### 3.2 Managed Element Fragment

Managed object classes in managed element fragment are presented in Figure 2. The definition(s) of the managed object class(es) are specified as follows:

#### 3.2.1 Equipment

equipment MANAGED OBJECT CLASS  
DERIVED FROM "Recommendation X.721: 1992" msg;  
CHARACTERIZED BY  
equipmentPackage PACKAGE  
BEHAVIOUR  
equipmentDefinitionMsg  
DEFINED AS

"The equipment object class is a class of managed objects that represent physical components of a managed element, including replaceable components. An instance of this object class is present in a single geographic location. An equipment may be tested while under repair, thereby creating a containment relationship. The equipment type shall be identified by the identifying this object class. Either the name of the sub-class or an instance may be used for identifying the equipment type."

When the attribute value change notification package is present, the networkValueChange notification defined in Recommendation X.721 shall be emitted when the value of one of the following attributes changes: alarm status, affected object list, user label, version, location name and current problem list. Because the above attributes are all in conditional packages, the behavior for emitting the attribute value change notification applies only when the corresponding conditional packages are present in the managed object. When the state change notification package is present, the stateChangeNotification defined in Recommendation X.721 shall be emitted if the value of administrative state or operational state changes (when the administrative/OperationalStates conditional package is present). "

ATTRIBUTES  
equipment GET;  
equipment GET;  
"

【図12】

## GDMO定義の一部を示す図(2)

**CONDITIONAL PACKAGES**  
 objectDetectionNotificationPackage PRESENT IF "the objectDetection and objectDetection notification defined in Recommendation X.721 are supported by an instance of this class".  
 attributeValueChangeNotificationPackage PRESENT IF "the attributeValueChange notification defined in Recommendation X.721 is supported by an instance of this class".  
 stateChangeNotificationPackage PRESENT IF "the stateChange notification defined in Recommendation X.721 is supported by an instance of this class".  
 stateChangeOperationalStatePackage PRESENT IF "an instance supports it".  
 objectObjectPackage PRESENT IF "an instance supports it".  
 equipmentEquipmentAlarmPackage PRESENT IF "the equipmentAlarm notification defined in Recommendation X.721 is supported by an instance of this class".  
 managementAlarmPackage PRESENT IF "the managementAlarm notification defined in Recommendation X.721 is supported by an instance of this class".  
 stateChangeAlarmPackage PRESENT IF "the stateChangeAlarm notification defined in Recommendation X.721 is supported by an instance of this class".  
 processEventPackage PRESENT IF "the processEvent notification defined in Recommendation X.721 is supported by an instance of this class".  
 mailObjectPackage PRESENT IF "an instance supports it".  
 vendorObjectPackage PRESENT IF "an instance supports it".  
 vendorPackage PRESENT IF "an instance supports it".  
 locationObjectPackage PRESENT IF "an instance supports it".  
 communicationPackage PRESENT IF "an instance supports it".

REGISTERED AS (asGuidedClass 2);

## 3.3.3 Managed Element

**ManagedElement** **MANAGED OBJECT CLASS**  
**DERIVED FROM** Recommendation X.721: ManagedElement  
**CHARACTERIZED BY**  
 managedElementPackage PACKAGE  
**BEHAVIOUR**  
**managedElementBehaviour** **BEHAVIOUR**  
**DEFINED AS**  
 "The ManagedElement object class is a class of managed objects representing physical objects or logical entities (other groups or parts) within the Recommendation X.721 environment. Managed elements may or may not perform notification functions. An managed element communicates with the manager over one or more standard Q interfaces for the purpose of being managed under control. A managed element can be deployed in any or may not be geographically distributed.  
 When the attribute value change notification package is present, the attributeValueChange notification defined in Recommendation X.721 shall be defined when the value of one of the following attributes changes: group, name, type, label, position, location name and current position. For the state change notification package, the behaviour for sending the state change notification applies only when the corresponding notification package is present in the managed object. When the state change notification package is present, the stateChange notification defined in Recommendation X.721 shall be defined if the value of administrative state or operational state or usage state changes."  
 ::  
**ATTRIBUTES**  
 managedElement GET,  
 Recommendation X.721: 1992: dynamic GET-REPLACE,  
 stateChange GET,

PTO 05-1619

Japanese Kokai Patent Application  
No. Hei 8[1996]-63483

INFORMATION ANALYSIS AND EDITING SYSTEM

Kenichi Fukuda

UNITED STATES PATENT AND TRADEMARK OFFICE  
WASHINGTON, D.C. JANUARY 2005  
TRANSLATED BY THE RALPH MCELROY TRANSLATION COMPANY

JAPANESE PATENT OFFICE  
PATENT JOURNAL (A)  
KOKAI PATENT APPLICATION NO. HEI 8[1996]-63483

Int. Cl. <sup>6</sup> :	G 06 F 17/30 17/27 G 06 F 15/40 15/38
Sequence Nos. for Office Use:	9194-5L 8420-5L
Filing No.:	Hei 6[1994]-194988
Filing Date:	August 19, 1994
Publication Date:	March 8, 1996
No. of Claims:	4 (Total of 11 pages; OL)
Examination Request:	Not filed

INFORMATION ANALYSIS AND EDITING SYSTEM  
[Joho kaiseki henshu shisutemu]

Inventor:	Kenichi Fukuda
Applicant:	000005223 Fujitsu Ltd.

[There are no amendments to this patent.]

Claims

1. An information analysis and editing system characterized in that it is equipped with a template definition extractor part for extracting template data for a part to be read from text data constructed using reserved words by means of syntactic analyses utilizing said reserved words and

a card type data generator part for recording the content of the aforementioned extracted template data into corresponding columns of card type data having standard item columns.

2. The information analysis and editing system described under Claim 1, characterized in that the text data comprise GDMO definition statements.

3. The information analysis and editing system described under Claim 2, characterized in that the template definition extractor part extracts multiple kinds of template data, and the card type data generator part records the aforementioned respective extracted template data into card type data for the respective templates.

4. An information analysis and editing system characterized in that it is equipped with a card type data editor part which takes the multiple card type data in Claim 3 as an input, runs a search within said card type data using the content of prescribed card type data as search keys, and collects pieces of information in item columns corresponding to 1 or 2 or more card type data found in order to generate new card type data.

#### Detailed explanation of the invention

[0001]

##### Industrial application field

The present invention pertains to an information analysis and editing system. More specifically, it pertains to an information analysis and editing system which extracts desired information from text data constructed using reserved words and edits it. In the case of the development of an OSI network management system based on the ISO international standard or a network management (to be managed) system based on the TMN of the ITU-T recommendations, compliance with GDMO definitions formulated by the ISO, ITU-T, or another organization is required. Thus, one who engages in the formulation of an interface specification and/or system designing needs to read and understand an enormous number of GDMO definitions in order to select a range of exhaustive (international) standard GDMO definitions to be implemented for a given device.

[0002]

##### Prior art

Figure 11 through Figure 13 are excerpts (1) through (3) showing portions of GDMO definitions used for conventional printing. In Figure 11, to ascertain the content of the definition "equipment" in an MO class, for example, "equipment MANAGED OBJECTCLASS" column under item 3.2.1 should be read. Here, "MANAGED OBJECTCLASS" shown in capital letters is a reserved word indicating the MO class, and "equipment" shown in lower-case letters indicates the name of the managing target object used in said definition statement.

[0003]

Furthermore, when the column for the reserved after subsequent [words] "DERIVED FROM" in the next stage is read to the end of the first segment ";" of said sentence structure, a



super class regarding “equipment” is defined by “top,” and the definition of “top” is described on a different page. Furthermore, other related detailed information is shown by the next reserved subsequent [words] “CHARACTERIZED BY.” For example, the name of the target of the reserved subsequent [words] “PACKAGE” is “equipmentPackage,” and its attributes (:ATTRIBUTES) are equipmentId (Note: Numbered as #1, #2, etc.) and replaceable (Note: Can be replaced).

[0004]

Furthermore, although it would be nice to ascertain “ACTIONS” and “NOTIFICATIONS (NOTIFICATIONS)” as a set in addition to “ATTRIBUTES” regarding “equipmentPackage,” references need to be made to other pages in the example given above. On the other hand, comments on “BEHAVIOUR” (:BEHAVIOUR) are embedded at the positions shown in the figure. However, it is not rare for “BEHAVIOUR” to be written on a far removed page.

[0005]

In Figure 12, “createDeleteNotificationsPackage” along with the names of various kinds of conditional packages are described in the column for the reserved words “CONDITIONAL PACKAGES” for the MO class “equipment.” To ascertain the contents (attributes, actions, notifications; etc.) of said respective conditional packages, reference needs to be made to other pages.

[0006]

“Notifications” of “createDeleteNotificationsPackage” is defined under item 4.10 in Figure 13. That is,  
createDeleteNotificationsPackage PACKAGE,  
NOTIFICATIONS,  
objectCreation (Note: data generation notification), and  
objectDeletion (Note: data deletion notification).

[0007]

In the past, the referential relationship between GDMO definitions was first traced manually with the help of the table of contents and the index of printed material in order to comprehend the meanings of the GDMO definitions. Subsequently, a range to be implemented to a device was selected based on the GDMO definitions.

[0008]

Problems to be solved by the invention

However, in reality, because

- ① an enormous number of GDMO definitions have been created,
- ② respective MO definitions are configured in multiple steps and are dispersed (For example, "BEHAVIOUR" is scattered around),
- ③ many complicated referential relationships are present among the MO definitions,
- ④ the accurate meaning of an MO is given indirectly based on its relationship to other MOs, and
- ⑤ MO definitions are not constructed in a format suitable for the selection of an implementation range, and a long time is required to interpret them correctly, so in order for one to become capable of selecting an implementation range, he/she needs to have a tremendous memory.

[0009]

Accordingly, in the past, there were various kinds of problems; for example,

- ⑥ GDMO definitions were very difficult to understand,
- ⑦ it was extremely difficult to select an implementation range, and
- ⑧ interface specifications had to be revised manually when GDMO definitions were revised.

[0010]

Although utilization of a document processor, such as an editor or a word processor, may appear feasible, said devices are capable only of searching a simple train of sentences; for example, "MANAGED OBJECT CLASS network" as the title of an MO definition and "MANAGED OBJECT CLASS network is an object which ..." as a comment within an MO definition are retrieved indiscriminately. The objective of the present invention is to present an information analysis and editing system by which desired information can be extracted from text data constructed using reserved words and can be edited for an intended objective easily.

[0011]

Means to solve the problems

The aforementioned problems can be solved using the configuration in Figure 1. That is, the information analysis and editing system in the present invention (1) is equipped with a template definition extractor part for extracting template data for a part to be read from text data constructed using reserved words by means of syntactic analyses utilizing said reserved words

and a card type data generator part for recording the content of the aforementioned extracted template data into corresponding columns of card type data having standard item columns.

[0012]

In addition, the information analysis and editing system in the present invention (4) is equipped with a card type data editor part which takes the multiple card type data in Claim 3 as an input, runs a search within said card type data using the content of prescribed card type data as search keys, and collects pieces of information in item columns corresponding to 1 or 2 or more card type data found in order to generate new card type data.

[0013]

Operation of the invention

In the information analysis and editing system of the present invention (1), template definition extractor part 1<sub>1</sub> extracts template data 72 for a part to be read from text data 71 constructed using reserved words by means of syntactic analyses utilizing said reserved words. Then, card type data generator part 1<sub>2</sub> records the content of the aforementioned extracted template data into corresponding columns of card type data 73 having standard item columns.

[0014]

Therefore, according to the present invention (1), because the template data for the part to be read within text data 71 with a complicated document structure are converted into card type data 73 having standard item columns for output, collection of the necessary information can be automated, and a database suitable for an information search can be constructed easily. Preferably, the text data comprise GDMO definition statements.

[0015]

Therefore, complicated GDMO definitions and referencing can be processed by a computer to contribute greatly to the reduction of interface specification creation work for a network management (to be managed) system. In addition, preferably, template definition extractor part 1<sub>1</sub> extracts multiple kinds of template data 72<sub>1</sub> through 72<sub>m</sub>, and card type data generator part 1<sub>2</sub> records aforementioned respective extracted template data 72<sub>1</sub> through 72<sub>m</sub> as card type data 73<sub>1</sub> through 73<sub>m</sub> for the respective templates.

[0016]

Therefore, even when respective template data 72<sub>1</sub> through 72<sub>m</sub> have unique structures as is evident with GDMO definitions, for example, an efficient database can be constructed. In

addition, in the information analysis and editing system of the present invention (4), card type data editor part 1<sub>3</sub> takes multiple card type data 73 in Claim 3 as an input, runs a search within said card type data using the content of prescribed card type data as search keys, and collects pieces of information in item columns corresponding to 1 or 2 or more card type data found in order to generate new card type data 74.

[0017]

Therefore, according to the present invention (4), various kinds of referential relationships contained in the original text data can be traced based on a database of GDMO definitions obtained in Claim 3, for example, in order to create a detailed data lookup table, whereby a range to be implemented to a device can be understood easily to facilitate a choice by looking at said [table].

[0018]

#### Application example

An application example in accordance with the present invention will be explained below according to attached figures. Furthermore, the same symbols indicate the same parts or their equivalents throughout the figures. Figure 2 is a block diagram of the information analysis and editing system of the application example. In the figure, 1 represents a CPU, 2 represents a ROM containing a program in order for CPU 1 to execute the processing in Figure 3 through Figure 5, 3 represents a display (DSP), 4 represents a disk device (DSK), 5 represents a keyboard (KBD), 6 represents a pointing device (PD) such as a mouse, 7 represents a RAM, 71 represents a text data storage area, 72 represents a template definition data storage area, 73 represents a card type data storage area, 74 represents an editing result storage area, 75 represents other area, 8 represents a printer (PRN), and 9 represents a common bus for CPU 1.

[0019]

CPU 1 works with ROM 2 to realize various functions for template definition extractor part 1<sub>1</sub>, card type data generator part 1<sub>2</sub>, and card type data editor part 1<sub>3</sub> of Figure 1. RAM 7 stores portions of text data 71<sub>1</sub> through 71<sub>k</sub>, template definition data 72<sub>1</sub> through 72<sub>m</sub>, card type data 73<sub>1</sub> through 73<sub>m</sub>, and editing result data 74<sub>1</sub> through 74<sub>n</sub>, and the rest is stored in the database of disk device 34.

[0020]

Figure 3 is a flow chart for template definition extraction processing in the application example. During the template definition extraction processing, GDMO text data are input, and

text data (template definition data) for a part to be read are queried by means of syntactic analyses utilizing said reserved words. GDMO text data contain various kinds of text data, such as communication network related text data 71<sub>1</sub> and computer network related text data 71<sub>2</sub>. The text data are constructed using reserved words as shown in Figure 11. However, columns described along with item numbers, for example,

### “3.2 Managed Element Fragment

Managed object classes in ... as follows:

#### 3.2.1 Equipment.”

are unique to a given printed material, and they are not included in the GDMO text data.

[0021]

GDMO text data are input in Step S1. A part of the template definition statements (for example, columns equivalent to the columns under item 3.2.1 in Figure 11 and Figure 12) are read in Step S2. In Step S3, reserved words are checked from the beginning in order to determine whether they are of the designated type (that is, template definition of the part to be read) or not. Designation type includes MO (MANAGED OBJECTCLASS), packages (PACKAGE, CONDITIONALPACKAGES), attributes (ATTRIBUTES), actions (ACTIONS), behaviour (BEHAVIOUR), and notifications (NOTIFICATIONS).

[0022]

If they are of the designated type, syntactic analyses are applied to the content based on the reserved words in the text data, symbol “;”, and the hierarchical structure of definition statements, and bound symbols are inserted in order to extract each template definition datum. If they are not of the designated type, the processing in Step S4 is skipped. In Step S5, whether there are any other template definitions embedded in the aforementioned part of the template definition statements or not is checked, and if any exist whether they are of the designated type or not is determined in Step S6. If they are of the designated type, syntactic analyses are applied to the content in Step S7, and bound symbols are inserted in order to extract new template definition data. If they are not of the designated type, the processing in Step S7 is skipped.

[0023]

When the extraction of the necessary template definitions is completed by checking all the embedded definitions in said manner, whether another part of template definitions is present or not is checked in Step S8. If present, [the process] returns to Step S2 to repeat the aforementioned processing. If not present, said processing is skipped. As a result, various kinds

of multiple template definition data  $72_1$  through  $72_m$  ranging from MO to ATTRIBUTES in Figure 1 are extracted individually.

[0024]

Figure 4 is a flow chart for card type data generating processing in the application example. During the card type data generating processing, the contents of the aforementioned various kinds of template definition data extracted above are written into the corresponding item columns. Examples of card data corresponding to the MO template are shown in Figure 6. Said card type data are provided with standard item columns (write fields), such as MO class, superclass, package, and conditional package. Similarly, Figure 7 and Figure 8 show examples of card type data with different write fields which correspond to package template and behaviour template, respectively.

[0025]

In Step S11, data on 1 template definition are read. Corresponding card type data (blank data) are generated in Step S12. A card type data write field is selected in Step S13, and data (for example, target name) corresponding to the applicable part of the template definition are written into said write field in Step S14. The aforementioned processing is applied to all write fields in Step S15. Then, in Step S16, the aforementioned processing is applied to all template definition data. As a result, multiple card type data  $73_1$  through  $73_m$  are generated for each template in Figure 1.

[0026]

Figure 6(A) shows an example display in which the MO class “equipment” in Figure 11 is unfolded as card type data, wherein “top” is written in the super class write field, and the target names of the various kinds of conditional packages in Figure 12 are written in the conditional package write field. An example of other card type data (MO class “broadbandATMSwitchingElement”) generated in the same manner is shown in Figure 6(B).

[0027]

Here, the part which cannot be seen in the frame can be displayed on the screen when the arrows provided on the right are clicked for scrolling. In addition, the upper right part of the card indicates that this card is the 39th card of a total of 298 MO class cards. In addition, although attribute names, action names, and notification names described in package template definitions are not shown on this MO class card, separate package template corresponding cards are created

as shown in Figure 7(A) and (B) with information regarding those definitions. In addition, Figure 8 shows an example of a card corresponding to a behaviour template.

[0028]

Figure 5 is a flow chart for card type data editing processing in the application example. For example, during the card type data editing processing (1), the MO class template definition corresponding card type data 73<sub>1</sub> and the package template definition corresponding card type data 73<sub>2</sub> are input, and the referential relationship between the cards is checked using the name of a prescribed package in order to combine them to generate new card type data 74<sub>1</sub>.

[0029]

That is, in Step S21, for example, the content of the superclass field of the MO class card is read into a column of list (LST) 1 provided in the other area in RAM 7. Similarly, the content of the package field of the MO class card is read into LST2 in Step S22. In Step S23, a package name is removed from LST2, and a search is run in package template corresponding card type data 73<sub>2</sub> using said [package name] as a search key. Then, once a search result is obtained, the contents of the attribute field, the action field, and the notification field of the package card are read into list (lst) 1, lst2, and lst3 provided in RAM 7, respectively, in the same manner in Step S24. In Step S25, the aforementioned processing is repeated with all package names in LST2.

[0030]

In Step S26, the content of the conditional package field of the MO class card is read into LST 3 in RAM 7. In Step S27, 1 package name is taken from LST3, and a search is run in package template corresponding card type data 73<sub>2</sub> using said [package name] as a search key. Then, once a search result is obtained, the contents of the attribute field, the action field, and the notification field of the package card are read (linked) into lst1, lst2, and lst3 in RAM 7, respectively, in the same manner in Step S28. In Step S29, the aforementioned processing is repeated with all package names in LST3.

[0031]

In Step S30, new card type data 74<sub>1</sub> having a superclass field, an attribute field, an action field, and a notification field are generated. In Step S31, the content of LST1 in RAM 7, the content of lst 1, the content of lst2, and the content of lst3 are read into the superclass field, the attribute field, the action field, and the notification field of card type data 74<sub>1</sub>, respectively.

[0032]

An example of card type data generated as a result of the editing is shown in Figure 9. Card type data 74<sub>1</sub> generated through card type editing processing (1) show the relationship among the name of the MO class, attributes which belong to said class, action, and notifications in a orderly easy-to-understand manner, presenting orderly meaningful information to a worker who tries to comprehend the GDMO definitions.

[0033]

In Figure 10, the card in Figure 9 is taken as an input, a search is run in card type data 73 using the superclass name as a search key, and a result is obtained by combining the attributes, the actions, and the notifications of multiple cards as new card type data 74<sub>2</sub>. Said card type data 74<sub>2</sub> show the kinds of attributes, actions, and notifications which should be attained when an MO class instance is created.

[0034]

When card type data 73 (or 74, depending on the case) in the application example are utilized in the aforementioned manner, various kinds of data editing can be achieved to meet intended purposes. As a result, a designer of the device can determine a device implementation range easily by selecting contents of card type data 74 in Figure 9 or Figure 10. Furthermore, although GDMO definition text data were used for input in the aforementioned application example, the present invention can be applied to other kinds of text data which are structured using reserved words in the same manner.

[0035]

In addition, although a specific application example suitable for the aforementioned invention was described, the configuration and the program processing can be changed in a variety of ways without going beyond the idea of the present invention as a matter of course.

[0036]

#### Effect of the invention

As described above, according to the present invention (1), necessary information can be collected automatically from text data having a complicated sentence structure, and a database suitable for an information search can be constructed easily. In addition, according to the present invention (4), a detailed data lookup table can be created by tracing various kinds of referential relationships in the original text data based on the aforementioned database, and a device implementation range can be comprehended and selected easily by looking at said [table].



[0037]

Therefore, the use of the present system for the presentation of GDMO definitions contributes greatly in terms of a reduction in the interface specification creation work for a network management (to be managed) system.

#### Brief description of the figures

Figure 1 is a diagram for explaining the principles of the present invention.

Figure 2 is a block diagram of the information analysis and editing system of an application example.

Figure 3 is a flow chart for template definition extraction processing in the application example.

Figure 4 a flow chart for card type data generating processing in the application example.

Figure 5 a flow chart for card type data editing processing in the application example.

Figure 6 shows diagrams (1) showing card type data in the application example.

Figure 7 shows diagrams (2) showing card type data in the application example.

Figure 8 is a diagram (3) showing card type data in the application example.

Figure 9 is a diagram (1) showing card type data obtained as a result of editing in the application example.

Figure 10 is a diagram (2) showing card type data obtained as a result of editing in the application example.

Figure 11 is an excerpt (1) showing a portion of GDMO definitions.

Figure 12 is an excerpt (2) showing a portion of GDMO definitions.

Figure 13 is an excerpt (3) showing a portion of GDMO definitions.

#### Explanation of symbols

- 1<sub>1</sub>.            template definition extractor part
- 1<sub>2</sub>.            card type data generator part
- 1<sub>3</sub>.            card type data editor part
- 71<sub>1</sub>-71<sub>k</sub>.       text data
- 72<sub>1</sub>-72<sub>m</sub>.       template definition data
- 73<sub>1</sub>-73<sub>m</sub>.       card type data
- 74<sub>1</sub>-74<sub>n</sub>.       resulting edited data

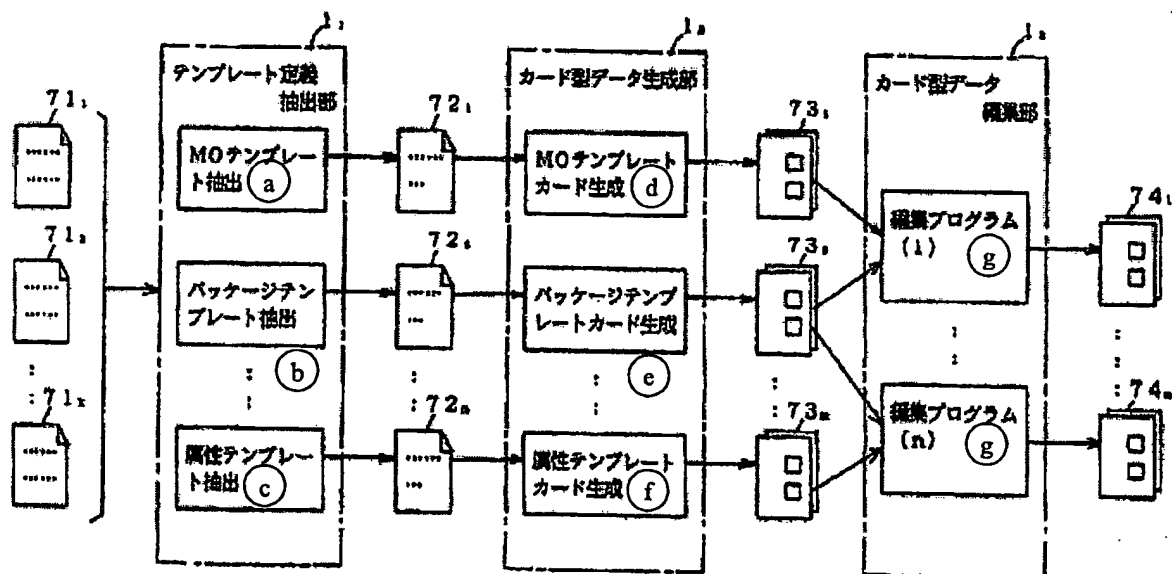


Figure 1. Diagram for explaining the principles of the present invention

- Key:
- a Extraction of MO template
  - b Extraction of package template
  - c Extraction of attribute template
  - d Generation of MO template card
  - e Generation of package template card
  - f Generation of attribute template card
  - g Editing program
  - 1<sub>1</sub> Template definition extractor part
  - 1<sub>2</sub> Card type data generator part
  - 1<sub>3</sub> Card type data editor part

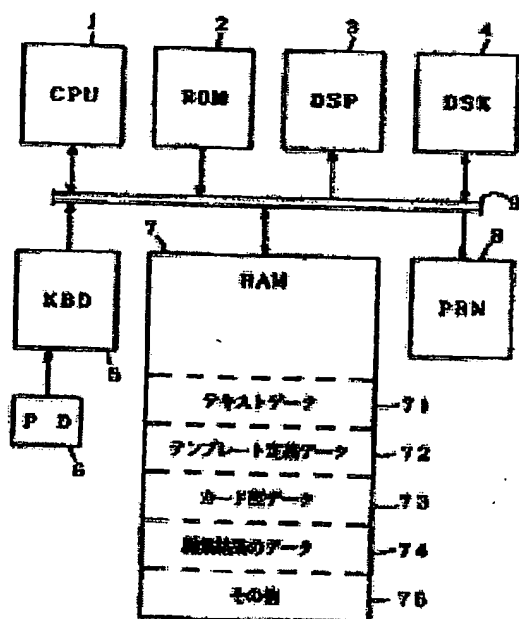


Figure 2. Block diagram of information analysis and editing system of application example

- Key: 71 Text data  
 72 Template definition data  
 73 Card type data  
 74 Resulting edited data  
 75 Other

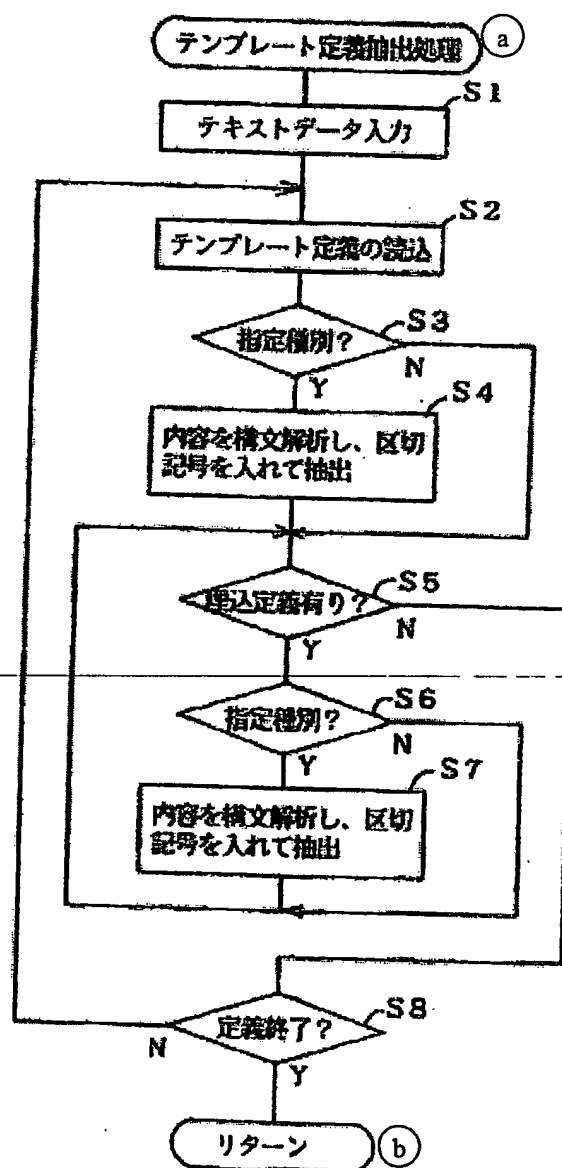


Figure 3. Flow chart for template definition extraction processing in an application example

- Key:
- a     Template definition extraction processing
  - b     Return
  - S1    Input text data
  - S2    Read template definition
  - S3, S6 Designated type?
  - S4, S7 Apply syntactic analyses to contents, and insert bound symbols for extraction
  - S5    Embedded definition present?
  - S8    Definitions completed?

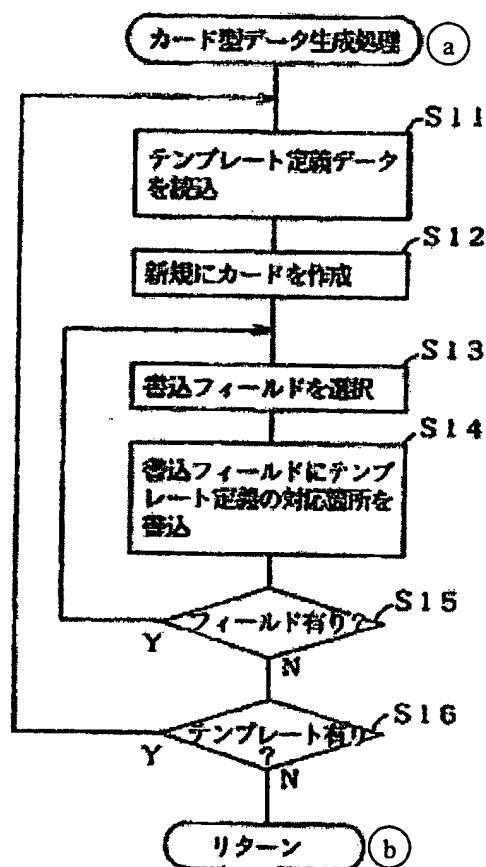


Figure 4. Flow chart for card type data generating processing in an application example

Key:	a	Card type data generating processing
	b	Return
	S11	Read template definition data
	S12	Create new card
	S13	Select write field
	S14	Write corresponding part of template definition into write field
	S15	Field present?
	S16	Template present?

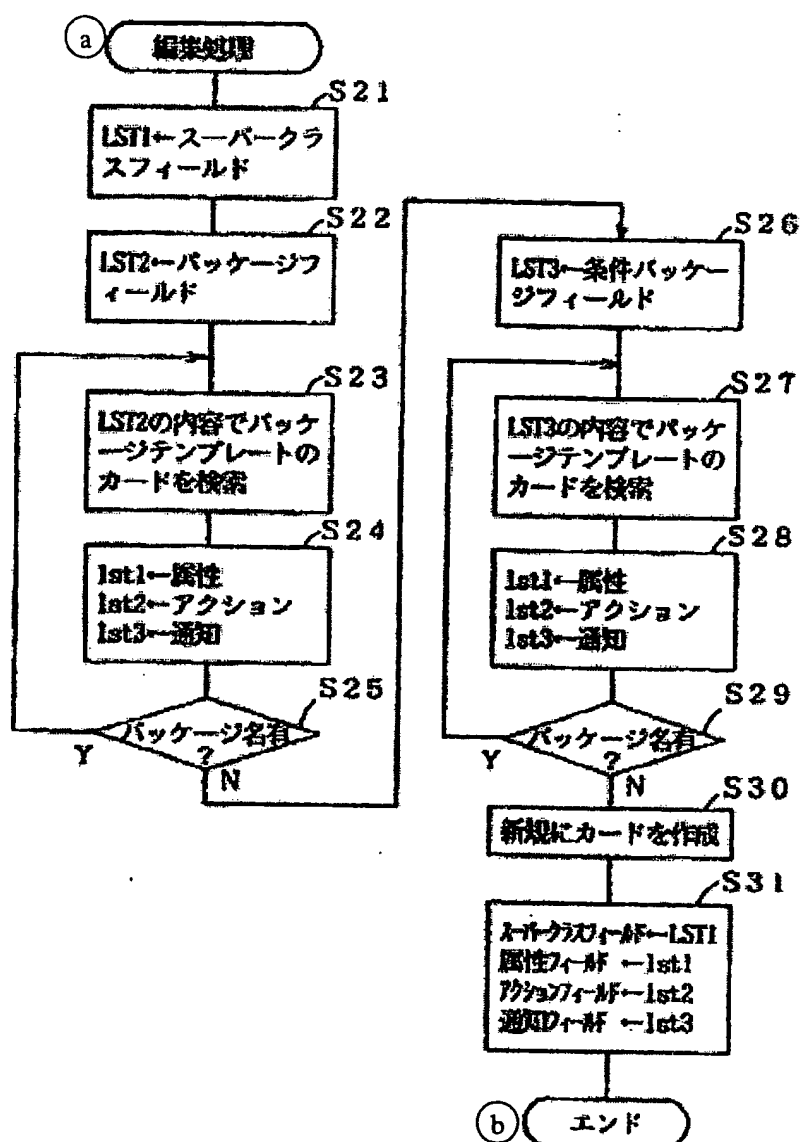


Figure 5. Flow chart for card type data editing processing in an application example

Key: a	Editing processing
b	End
S21	LST1 ← superclass field
S22	LST2 ← package field
S23	Run search in package template using content of LST2
S24, S28	lst1 ← attribute
	lst2 ← action
	lst3 ← notification
S25, S29	Package name present?
S26	LST3 ← conditional package field
S27	Run search in package template using content of LST3

S30 Create new card  
 S31 Superclass field ← LST1  
 Attribute field ← lst1  
 Action field ← lst2  
 Notification field ← lst3

① クラス 30/298

スーパークラス

②

③ パッケージ

④ 条件パッケージ

createChildOfFirstPackage  
 createChildOfSecondPackage  
 createChildOfThirdPackage  
 createChildOfFourthPackage  
 createChildOfFifthPackage  
 createChildOfSixthPackage  
 createChildOfSeventhPackage  
 createChildOfEighthPackage  
 createChildOfNinthPackage  
 createChildOfTenthPackage

(A)

① クラス 40/298

スーパークラス

②

③ パッケージ

④ 条件パッケージ

createChildOfFirstPackage  
 createChildOfSecondPackage  
 createChildOfThirdPackage  
 createChildOfFourthPackage  
 createChildOfFifthPackage  
 createChildOfSixthPackage  
 createChildOfSeventhPackage  
 createChildOfEighthPackage  
 createChildOfNinthPackage  
 createChildOfTenthPackage

(B)

Figure 6. Diagram (1) showing card type data in an application example

Key: 1 Class  
 2 Superclass  
 3 Package  
 4 Conditional package

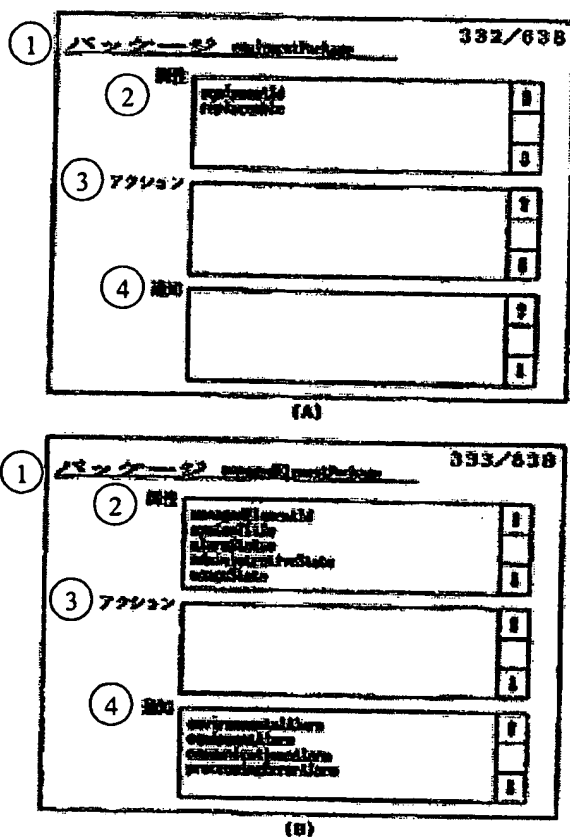


Figure 7. Diagram (2) showing card type data in an application example

Key: 1 Package  
2 Attribute  
3 Action  
4 Notification

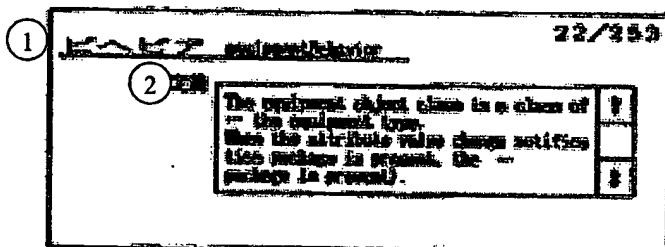


Figure 8. Diagram (3) showing card type data in an application example

Key: 1 Behaviour  
2 Definition



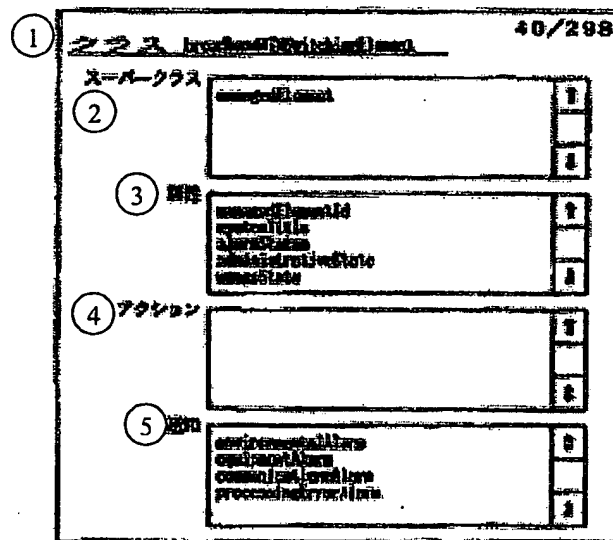


Figure 9. Diagram (1) showing card type data obtained as a result of editing in the application example

- Key: 1 Class  
2 Superclass  
3 Attribute  
4 Action  
5 Notification

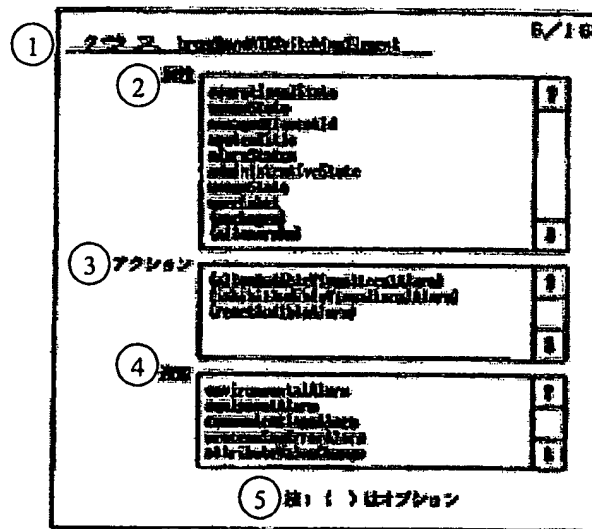


Figure 10. Diagram (2) showing card type data obtained as a result of editing in the application example

Key: 1      Class  
 2      Attribute  
 3      Action  
 4      Notification  
 5      Note: ( ) is optional





```

ATTACHED
  CHARACTERISTICS:
    4.7 Characteristic Information
      ATTACHED
        CHARACTERISTICS:
          4.8 Class Overview
            ATTACHED
              ATTACHED
                CHARACTERISTICS:
          4.9 Class Trail
            ATTACHED
              ATTACHED
                CHARACTERISTICS:
          4.10 Class: Data Transformation
            ATTACHED
              ATTACHED
                CHARACTERISTICS:
          4.11 Class: Connection Point
            ATTACHED
              ATTACHED
                CHARACTERISTICS:
          4.12 CTP: Access
            ATTACHED
              ATTACHED
                CHARACTERISTICS:

```

Figure 13. Excerpt (3) showing a portion of GDMO definitions

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**